



**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE**

# **Zookeeper**

## **Laboratorium**

Systemy Rozproszone

**Piotr Nawrocki**

**Wydział Informatyki AGH**

## Ćwiczenie 1

- W katalogu `apache-zookeeper-3.8.2-bin\conf\` zmienić nazwę pliku `zoo_sample.cfg` na `zoo.cfg`
- W `apache-zookeeper-3.8.2\bin\`
  - Uruchomić serwer Zookeeper - `zkServer.cmd`
  - Uruchomić klienta Zookeeper - `zkCli.cmd -server 127.0.0.1:2181`

### Linux/macOS

- Uruchomić serwer Zookeeper - `zkServer.sh start-foreground`
- Status serwera Zookeeper - `zkServer.sh status`
- Uruchomić klienta Zookeeper - `zkCli.sh -server 127.0.0.1:2181`

## Ćwiczenie 1 - podstawowe operacje

- W kliencie wydać komendę *help* – lista wszystkich dostępnych komend
- Wyświetlenie listy *znode*: ***ls / (lub ls -R /)***
- Każdy *znode* może zawierać dane (tak jak plik, domyślny limit danych 1 MB) oraz potomków (podobnie jak katalog)
- Stworzenie przykładowego pustego *znode* (będącego parent 'directory'): ***create /zk-demo Pierwszy***
- Stworzenie potomka tego *znode*: ***create /zk-demo/my-node Moje\_dane***
- Wyświetlenie listy *znode*: ***ls /***
- Wyświetlenie listy *znode*: ***ls /zk-demo***

## Ćwiczenie 1 - podstawowe operacje

- Pobranie informacji o znode: **get -s /zk-demo/my-node**

```
[zk: 127.0.0.1:2181(CONNECTED) 20] get -s /zk-demo/my-node
```

*Moje\_dane*

**cZxid** = 0x600000003 // transaction id stworzenie znode w Zookeeper

**ctime** = Wed Apr 15 18:32:16 CEST 2015 //czas utworzenia znode

**mZxid** = 0x600000004 //transaction id ostatniej zmiany znode w Zookeeper

**mtime** = Wed Apr 15 18:36:28 CEST 2015 //czas ostatniej modyfikacji znode

**pZxid** = 0x600000003 //transaction id ostatniej modyfikacji (add/delete)

*potomka*

**cversion** = 0 // liczba zmian w potomku znode

**dataVersion** = 1 //liczba zmian danych w znode

**aclVersion** = 0 // liczba zmian w ACL (listach kontroli dostępu) znode

**ephemeralOwner** = 0x0 //session ID właściciela znode typu ephemeral

**dataLength** = 9 //długość danych w polu danych znode

**numChildren** = 0 //liczba potomków w znode

## Ćwiczenie 1 - podstawowe operacje

- Zmiana informacji w znode: ***set /zk-demo/my-node dane***
- Pobranie informacji o znode: ***get -s /zk-demo/my-node***
- Skasowanie znode: ***delete /zk-demo/my-node***
- Skasowanie znode z potomkami: ***deleteall /zk-demo)***

## Ćwiczenie 2 – znode sequential i ephemeral

- Istnieje możliwość stworzenia dwóch specjalnych typów znode: sequential (*create -s*) i ephemeral (*create -e*).
- Dla opcji sequential tworzone są znode posiadające (unikalny) numeryczny sufiks
- Stworzenie znode typu sequential: **create -s /zk-demo/sequential dane1**
- Stworzenie znode typu sequential: **create -s /zk-demo/sequential dane2**
- Wyświetlenie listy znode **ls /zk-demo**

## Ćwiczenie 2 - znode sequential i ephemeral

- Stworzenie znode typu ephemeral (emferyczny):  
**create -e /zk-demo/ephemeral dane**
- Tego typu znode działa do momenty wyłączenia klienta, po tym jest kasowany
- Wyjście z klienta i ponowne uruchomienie, na liście znode nie ma **ephemeral**
- Wyświetlenie listy znode **ls /zk-demo** -> są tylko dwa znode typu sequential

## Ćwiczenie 3 - watches

- Istnieje możliwość monitorowania stanu znode (i jego potomków) poprzez mechanizm *watches*
- Tworzenie *znode*: ***create /zk-demo/watch-this data***
- Ustawienie obserwacji *znode*: ***get -s -w /zk-demo/watch-this*** lub ***stat -w /zk-demo/watch-this***
- Monitorowanie *znode* działa tylko raz (jak nastąpi zmiana). W celu dalszego monitorowania trzeba ponownie ustawić obserwacje *znode*



## Ćwiczenie 4 - replikacja

- Zookeeper umożliwia replikację danych (Replicated ZooKeeper) pomiędzy wieloma serwerami – zmiany znode na jednym serwerze widoczna jest na pozostałych
- Liczba serwerów Zookeepera w trybie replikacji powinna być nieparzysta (wybór lidera)
- Grupa serwerów (z replikacją) dla tej samej aplikacji nazwa się *quorum*

## Ćwiczenie 4 - replikacja

- Trzeba przygotować trzy pliki konfiguracyjne i umieścić je w katalogu conf
- Każdy plik powinien zawierać:  
*dataDir=/tmp/zookeeper/zk1 (inne dla każdej konfiguracji)*  
*clientPort=2181 (inny port dla każdej konfiguracji)*  
*server.1=localhost:2888:3888 (2888 – komunikacja, 3888 – wybór lidera)*  
*server.2=localhost:2889:3889*  
*server.3=localhost:2890:3890*
- W każdym katalogu (zk1, ...) trzeba stworzyć plik tekstowy o nazwie *myid* (bez rozszerzenia *.txt*) podający numer serwera (**1**, ...)

## Ćwiczenie 4 - replikacja

- Trzeba uruchomić trzy serwery: **zkServer2.cmd** \ **zoo1.cfg ... (zoo2.cfg, zoo3.cfg)**
- Skrypt **zkServer2.cmd** dostępny na Moodle
- Dla systemu Linux/MacOS: zkServer.sh --config dir\_conf1 (dir\_conf2, dir\_conf3) start-foreground
- Trzeba uruchomić aplikacje klienta dla każdego z serwerów
- Na pierwszy serwerze (1) trzeba dodać **zn** i na pozostałych sprawdzić stan serwera (**komenda ls /**)
- Na trzecim serwerze (3) trzeba skasować **zn** i na pozostałych sprawdzić stan serwera (**komenda ls /**)
- Sprawdzić który serwer jest liderem ?

## Ćwiczenie 4 - replikacja

- Dla systemu Windows:
  - Wydać komendę *telnet localhost nr. portu (np. 2181)*
  - *Wydać komendę srvr* - informacje na temat serwera (może nie być widać wpisywanego tekstu)
  - Zaobserwować parametr Mode
    - Mode: follower
    - Mode: leader
  - localhost:8080/commands/leader
- Dla systemu Linux:
  - zkServer.sh --config dir\_conf1 status
  - zkServer.sh --config dir\_conf2 status
  - localhost:8080/commands/leader